Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2010-043
CBCL-291

September 9, 2010

Reliably Detecting Connectivity using Local Graph Traits

Alejandro Cornejo and Nancy Lynch

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Reliably Detecting Connectivity using Local Graph Traits

Alejandro Cornejo and Nancy Lynch

Massachusetts Institute of Technology,
Cambridge MA 02139-4307, USA

Abstract. Local distributed algorithms can only gather sufficient information to identify local graph traits, that is, properties that hold within the local neighborhood of each node. However, it is frequently the case that global graph properties (connectivity, diameter, girth, etc) have a large influence on the execution of a distributed algorithm.
This paper studies local graph traits and their relationship with global graph properties. Specifically, we focus on graph k-connectivity. First we prove a negative result that shows there does not exist a local graph trait which perfectly captures graph k-connectivity. We then present three different local graph traits which can be used to reliably predict the k-connectivity of a graph with varying degrees of accuracy.
As a simple application of these results, we present upper and lower bounds for a local distributed algorithm which determines if a graph is k-connected. As a more elaborate application of local graph traits, we describe, and prove the correctness of, a local distributed algorithm that preserves k-connectivity in mobile ad hoc networks while allowing nodes to move independently whenever possible.

1 Introduction

The t-neighborhood of a node u of a graph G, is the induced subgraph of G consisting of all vertices at distance at most t from u, and all edges connecting two such vertices. A graph trait is a pair (t, T) where t is a function from the positive integers to the positive integers, and T is a predicate over a graph. A graph trait (t, T) is satisfied by a graph G on n vertices, if the $t(n)$-neighborhood of every node of G satisfies T. A graph trait is local if t is a constant.

Our motivation for studying local graph traits comes from the classical synchronous distributed system model. In this model, each node of an undirected graph G is occupied by a processor. The system progresses in synchronous lockstep rounds, and at each round a process can send a message to its neighbors, receive messages, and perform local computation. Observe that after running for t rounds, the knowledge of a process is limited to learning about all nodes at distance at most t, as well as the edges present between these nodes (i.e. its t-neighborhood). Since we do not restrict either the amount of local computation or the message size, it follows that after $O(diameter(G))$ rounds, every process can acquire complete knowledge of the graph and can compute any function of

G. Therefore, distributed algorithms whose runtime is independent of the diameter of the network are especially interesting. Awerbuch et al. [1] defined a *local* algorithm as one whose runtime is significantly smaller than n for any possible diameter of the network [1]. Local distributed algorithms can only learn their local neighborhood, and therefore they are limited to observing local graph traits.

Despite the fact that local distributed algorithms are limited to observe local graph traits, it is often the case that global graph properties have a great influence on the execution of a distributed algorithm. For example, the chromatic number of a graph is a lower bound on the number of rounds required for every node to broadcast once without colliding with its neighbors. Similarly, in algorithms which require coordination, the connectivity of a graph is an upper bound on the fault-tolerance of an algorithm, since higher connectivity implies more nodes can fail without disconnecting the graph.

Given the effects that graph properties, both local and global, have on the execution of distributed algorithms, it is not surprising that studying the relationship between local graph traits and global graph properties is a fruitful direction for proving upper and lower bounds on local distributed algorithms. This was first observed in the seminal work of Linial [11], who used an elegant construction relying on t-neighborhood graphs to prove that any distributed algorithm that finds a maximal independent set in a cycle must take at at least $\Omega(\log^* n)$ rounds.

However, the study of the relationship of local graph traits and global graph properties dates further back. In 1983, Wigderson [13] showed that if a graph is locally k-chromatic, then it has a chromatic number of $O(\sqrt{kn})$. Even earlier, in 1952, Dirac [6] proved that if G has at least three vertices, and all nodes have degree at least $n/2$, then G is Hamiltonian. In the same vein, we study local graph traits which imply global graph k-connectivity.

Paraphrasing the formal definition given in Section 2, the connectivity of a graph G, denoted $\kappa(G)$, is the size of the smallest set of vertices whose removal disconnects the graph. Although a complete graph on n vertices cannot be disconnected by removing vertices, by convention its connectivity is $n-1$. We say a graph G is k-connected if $\kappa(G) \geq k$. In Section 3, we show that there does not exist a local graph trait that characterizes a k-connected graph. More precisely, we prove that for any constant $k > 0$ there does not exist a local graph trait (t, T) such that a graph G satisfies (t, T) if and only if G is k-connected. We show a similar result holds even when considering only *simply connected* graphs. Namely, there does not exist a local graph trait (t, T) such that a connected graph G satisfies satisfies (t, T) if and only if G is k-connected. These results hold even in the case of unit disk graphs.

Since its not possible to locally characterize the k-connectivity of a graph, in Section 4 we turn our attention to local graph traits that when satisfied imply

[1] We remark that the algorithms presented in this paper satisfy a more stringent notion of locality, since their runtime is constant and therefore independent of n or the diameter of the network. However, our impossibility results hold for the weaker notion of locality.

k-connectivity. Specifically, we describe three different local graph traits which are parametrized by k, and when fulfilled imply that the is graph k-connected.

As a simple application of these results, in Section 5 we present straightforward algorithmic implementations of the local traits described in Section 4 which yield constant time distributed algorithms to test for k-connectivity. We also describe a lower bound for distributed algorithms that reliably predict k-connectivity, which is derived directly from the impossibility results described in Section 3. As a more elaborate application, we show how to exploit the local graph traits presented, to extend the algorithm described in [4] to preserve k-connectivity in a mobile ad hoc network while allowing the agents of the network to move as freely as possible.

Most of the previous work on k-connectivity is in the field of topology control. Jorgic et al. [8] reported the experimental results of three different distributed algorithms to detect k-connectivity on random geometric graphs, but the paper lacks any formal guarantees. Czumaj and Zhao [5] presented a greedy centralized algorithm to construct a k-connected t-spanner with runtime $\tilde{O}(nk)$. Thurimella [12] described a distributed algorithm to identify sparse k-connected subgraphs that runs in $O(diameter(G) + \sqrt{n})$ time. Jia et al. [7] described a centralized algorithm to approximate the minimum power assignment while preserving k-connectivity. Similarly, Li and Hou [9] describe a distributed algorithm that given a k-connected graph finds a k-connected spanner. A preliminary version of the algorithmic counterpart of two of the local graph traits described in Section 4 was presented in [3].

2 Model

The communication network is modeled as a undirected graph. We use $G = (V, E)$ to denote an undirected graph, where V is the set of vertices, and E is the set of edges (two-element subsets of V). A pair of vertices $u, v \in V$ are neighbors if and only if $\{u, v\} \in E$. For ease of exposition we use the notation $E(G)$ (and $V(G)$) to denote the set of edges (and vertices) of a graph G. It is well known that most graph functions cannot be computed in anonymous networks, even for very simple graphs G. Hence, we define labeled graphs, denoted by a tuple (G, id), where $id : V \to I$ is an injective function that maps each vertex to a unique identifier. In mobile ad hoc networks it is often useful to assume processes know their own position. To this end, we consider two-dimensional Euclidean graphs, denoted by a tuple (G, p) (or (G, p, id) when considering labeled two-dimensional Euclidean graphs) where $p : V \to \mathbb{R}^2$ is a function that maps each vertex to a point in the Euclidean plane. A two-dimensional Euclidean graph (G, p) is a *unit disk graph* if there is an edge between two nodes if and only if they are at distance at most one, that is $E := \{\{u, v\} \mid \|p(u) - p(v)\| \leq 1\}$.

We consider a synchronous network model. Specifically, each node of an undirected graph G is occupied by a process. The system progresses in synchronous lock-step rounds. At each round a process can send a message to its neighbors, receive messages from its neighbors and perform local computation. If the graph

is labeled, we assume that at time zero the processor occupying node $v \in V(G)$ knows the identifier $id(v)$ of that node. Similarly, if the graph has an associated embedding (i.e. two-dimensional Euclidean graphs) we assume that at time zero the processor occupying node $v \in V(G)$ knows the position $p(v)$ of that node.

For a positive integer t we denote with $N^t[u]$ the closed t-neighbors of u, the set of vertices reachable by paths starting at u and of length at most t. Let $G^t(u)$ be the t-neighborhood of node u, the graph induced by the closed t-neighbors of u in G. When $t = 1$ we simply use $N[u]$ and $G(u)$ to denote the 1-neighbors and 1-neighborhood of node u respectively.

Since this model does not restrict the message size or the amount of local computation, after t rounds a process at vertex v can learn about its t-neighbors (including their unique ids and embedding when considering labeled Euclidean graphs), but it cannot learn about any node which is more than t hops away. In particular in unit disk graphs, after t rounds a process at vertex v can learn exactly its t-neighborhood $G^t(v)$. In general graphs, after t rounds a node can learn all edges between its t-neighbors, except for those edges whose endpoints are at distance exactly t. As shown in Section 5, this subtle difference between unit disk graphs and general graphs can be bridged by using an additional communication round to learn the t-neighborhood of a node in $t+1$ communication rounds.

In Section 3 and 4 we e study the relationship between local graph traits and global graph properties. A graph trait is a pair (t, T) where t is a function from the positive integers to the positive integers and T is a predicate over a graph, which (if applicable) can make use of the labeling or embedding of the graph. A graph G on n vertices satisfies a trait (t, T), if the $t(n)$-neighborhood of every vertex $v \in V(G)$ satisfies T. A graph trait is local if $t \in O(1)$; a graph trait is weakly-local if $t \in o(n)$.

A graph trait (t, T) implies a graph property P if any graph which satisfies (t, T) also satisfies P. Similarly, a graph property P implies a graph trait (t, T) if any graph which satisfies P also satisfies (t, T). A graph trait (t, T) characterizes a graph property P (or alternatively a graph property P is characterized by a graph trait (t, T)) if (t, T) implies P and P implies (t, T). Given the graph traits (t, T) and (t', T') which imply a graph property P, we say that (t, T) is *more accurate* than (t', T') with respect to P if every graph which satisfies (t', T') also satisfies (t, T), and there exists a graph which satisfies P and (t, T), but not (t', T').

In particular, the global graph property that we are concerned with is graph k-connectivity. A *vertex cut* C of a connected graph G is a set of vertices whose removal renders G disconnected. The *size* of a vertex cut C is the number of vertices $|C|$. A vertex cut is said to be a *minimum vertex cut* if it is a vertex cut of smallest size. The connectivity of a graph G, denoted by $\kappa(G)$, is the size of a smallest vertex cut of G. A complete graph on n vertices has no cuts at all, but by convention its connectivity is $n - 1$. We say a graph G is k-connected if $\kappa(G) \geq k$.

3 The Impossibility of Locally Characterizing Connectivity

In this section we show that it is impossible to characterize the k-connectivity of a graph using (weakly-)local graph traits. The results hold even when restricted to simply connected labeled unit disk graphs. As a warm up, we first show that there does not exist a weakly-local graph trait that characterizes simple connected graphs.

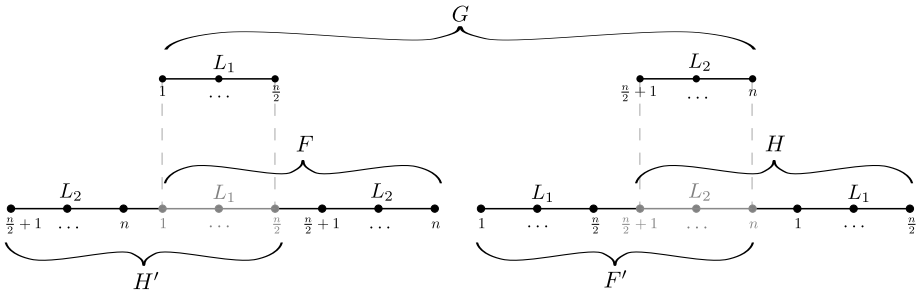

Fig. 1. All nodes are embedded in the horizontal axis. Neighboring nodes are 1 unit apart.

Theorem 1. *There does not exist a weakly-local graph trait (t, T) that characterizes a simply connected graph.*

Proof. Fix any local trait (t, T) which is implied by a simply connected graph. We will show that there exists a disconnected graph G which satisfies (t, T).

Since $t \in o(n)$ there exists a sufficiently large n such that $n > 4t(n)$, we consider graphs over the vertex set $V = \{1, \ldots, n\}$. Throughout the proof we assume all graphs are labeled using the same injective function. We group the vertices into two connected components L_1 and L_2. Component L_1 is a line graph of the first $\frac{n}{2}$ nodes, namely for each $i \in [1, \frac{n}{2} - 1]$ vertex i is connected with vertex $i + 1$. Component L_2 is a line graph of the remaining nodes, namely for each $i \in [\frac{n}{2} + 1, n - 1]$ node i is connected with node $i + 1$.

In the rest of the proof we describe how to connect L_1 and L_2 to produce a disconnected graph G and four connected graphs F, F', H and H'. We then show that since (t, T) is satisfied by the four connected graphs by assumption, it must be that G also satisfies (t, T).

Specifically, G is the disconnected graph made up of L_1 and L_2 with no additional edges. The graphs F and F' result from joining L_1 and L_2 with the edge $\left\{\frac{n}{2}, \frac{n}{2} + 1\right\}$, and the graphs H and H' result from joining L_1 and L_2 with the edge $\{n, 1\}$.

These graphs can be embedded as unit disk graph such that: (i) L_1 has the same embedding in F, G and H'. (ii) L_2 has the same embedding in F', G and H (cf. figure 1).

By assumption T is satisfied on the $t(n)$-neighborhood of every node in F, F', H and H'. To show that G satisfies the local trait (t, T), it suffices to show that every node has the same $t(n)$-neighborhood (including the labeling and embedding of the nodes) in G as it does in F, F', H or H'. We proceed by a case analysis on $i \in V$.

1. If $i \in [1, \frac{n}{4}]$ the $t(n)$-neighborhood of node i in G is a line graph with the nodes $\max(1, i - t(n)), \ldots, i, \ldots, i + t(n)$, which is the same $t(n)$-neighborhood of node i in F.
2. If $i \in [\frac{n}{4} + 1, \frac{n}{2}]$ the $t(n)$-neighborhood of node i in G is a line graph with the nodes $i - t(n), \ldots, i, \ldots, \min(i + t(n), \frac{n}{2})$, which is the same $t(n)$-neighborhood of node i in H'.
3. If $i \in [\frac{n}{2} + 1, \frac{3n}{4}]$ the $t(n)$-neighborhood of node i in G is a line graph with the nodes $\max(\frac{n}{2}, i - t(n)), \ldots, i, \ldots, i + t(n)$, which is the same $t(n)$-neighborhood of node i in H.
4. If $i \in [\frac{3n}{4} + 1, n]$ the $t(n)$-neighborhood of node i in G is a line graph with the nodes $i - t(n), \ldots, i, \ldots, \min(i + t(n), n)$, which is the same $t(n)$-neighborhood of node i in F'.

□

The previous theorem relies on the fact that if $t \in o(n)$ we can construct a large enough disconnected graph where every $t(n)$-neighborhood is indistinguishable from one in a connected graph. The same argument can be extended to show it is possible to construct a large enough disconnected graph whose $t(n)$-neighborhood is indistinguishable from a k-connected graph.

However, if we restrict our attention to characterizing the connectivity of *simply connected* graphs, the same argument no longer works. In particular in a graph which is connected but not k-connected, there exists a minimum vertex cut C of size $1 \leq |C| < k$. It is conceivable that the $t(n)$-neighborhood of a node $u \in C$ in the cut might not fulfill all the local traits implied by a k-connected graph. The following theorem rules out that possibility by showing that even when restricted to connected graphs, there does not exist a local graph trait that characterizes a k-connected graph.

Theorem 2. *For any constant $k > 0$ there does not exist a weakly-local graph trait (t, T) that characterizes k-connectivity of $\frac{k}{2}$-connected graphs.*

Proof. Let k be any positive constant, and fix any local trait (t, T) which is implied by k-connectivity. We will show that there exists a $\frac{k}{2}$-connected graph G which is not k-connected and satisfies (t, T).

Since k is a constant and $t \in o(n)$, then there exists a sufficiently large n such that $n = 2m \cdot k$ where $m > 4t(n)$, we consider graphs over the vertex set $V = \{1, \ldots, n\}$. We assume all graphs are labeled with the same injective

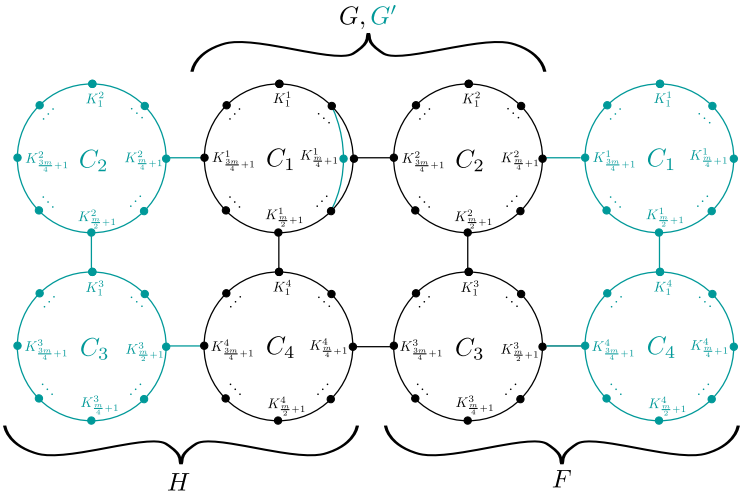

Fig. 2. Each point in the graph represents a clique of size $k/2$ embedded at that point, there is a line between cliques A and B if every node in clique A is connected to every node in clique B. The clique cycle C_i is formed by arranging the cliques uniformly around a circle at distance 1 from each other. To form F, H and G we arrange the clique cycles in a ring, where each clique cycle is at distance 1 from its neighboring clique cycle. To break the links between C_1 and C_2 in G' we "push" the nodes of $K^1_{\frac{m}{4}-1}$ some $\varepsilon > 0$ towards the center of C_1.

function. We partition the vertices V into four sets V_1, V_2, V_3 and V_4 each of size $mk/2$. Each vertex set V_i is partitioned further into m cliques $K^i_1, \ldots, K^i_m$, each of size $\frac{k}{2}$. In a slight abuse of notation we say cliques A and B are connected if every node in A is connected to every node in B. For each vertex set V_i we consider the clique cycle graph $C_i = \langle V_i, E_i \rangle$ formed by connecting clique K^i_j to clique $K^i_{j+1 \mod m}$ for each $j \in [1, m]$.

In the rest of the proof we describe how to connect these clique cycles to produce the k-connected graphs F, G and H, and the graph G' with connectivity $\frac{k}{2}$. We then argue that since (t, T) is satisfied by F, G and H by assumption, then it must also be satisfied by G', which completes the theorem.

To construct the graphs F, G and H we connect the clique cycles C_1, C_2, C_3 and C_4 in a ring. Specifically in G we connect cliques $K^1_{\frac{m}{4}+1}$ and $K^2_{\frac{3m}{4}+1}$, cliques $K^2_{\frac{m}{2}+1}$ and K^3_1, cliques $K^3_{\frac{3m}{4}+1}$ and $K^4_{\frac{m}{4}+1}$, and cliques K^4_1 and $K^1_{\frac{m}{2}+1}$. In F and H we connect cliques $K^1_{\frac{3m}{4}+1}$ and $K^2_{\frac{m}{4}+1}$, cliques $K^2_{\frac{m}{2}+1}$ and K^3_1, cliques $K^3_{\frac{m}{2}+1}$ and $K^4_{\frac{3m}{4}+1}$, and cliques K^4_1 and $K^1_{\frac{m}{2}+1}$. Finally, G' is the graph that results from removing the edges between C_1 and C_2 in G. Observe that to disconnect F, G or H we need to remove delete all the nodes of at least two cliques, and since each clique is of size $k/2$, it follows that these graphs are k-connected. Similarly, to disconnect G' it is sufficient and necessary to remove all the nodes of a single clique, and since each clique is of size $k/2$, it follows G' has connectivity $\frac{k}{2}$.

These graphs can be embedded as a unit disk graph such that: (i) The embedding of G and G' are identical except for the clique $K^1_{\frac{m}{4}+1}$. (ii) The clique cycles C_1 and C_4 have the same embedding in G' and H. (iii) The clique cycles C_2 and C_3 have the same embedding in G' and F (cf. figure 2).

By assumption G, F and H satisfy the local trait (t, T), hence the $t(n)$-neighborhood of every node satisfies T in each of these graphs. To show that G' satisfies the local trait (t, T) it suffices to show that $\forall i \in [1, 4], \forall j \in [1, m]$ every node $v \in K^i_j$ has the same $t(n)$-neighborhood in G' as it does in G, F or H.

Observe that G and G' only differ by the embedding of $K^1_{\frac{m}{4}+1}$ and by the presence (or lack thereof) of the edges between $K^1_{\frac{m}{4}+1}$ and $K^2_{\frac{3m}{4}+1}$. Therefore, for any node whose $t(n)$-neighborhood does not include a node from $K^1_{\frac{m}{4}+1}$ or $K^2_{\frac{3m}{4}+1}$, its $t(n)$-neighborhood is identical in G and G'. Moreover since $t(n) < m/4$ only a "few" nodes in C_1 and C_2 include a node from $K^1_{\frac{m}{4}+1}$ or $K^2_{\frac{3m}{4}+1}$ in their $t(n)$-neighborhood. Specifically, only a node v in clique K^1_j for $j \in [2, \frac{m}{2}]$ can include a node from $K^1_{\frac{m}{4}+1}$ in its $t(n)$-neighborhood. However, its $t(n)$-neighborhood cannot include a node from $K^1_{\frac{3m}{4}+1}$, and hence its $t(n)$-neighborhood is identical in G' and F. Similarly, only a node v in clique K^2_k for $k \in [\frac{m}{2} + 2, m]$ can include a node from $K^2_{\frac{3m}{4}+1}$ in its $t(n)$-neighborhood. However, its $t(n)$-neighborhood cannot include a node from $K^2_{\frac{m}{4}+1}$, and hence its $t(n)$-neighborhood is identical in G' and H. □

Given that it is impossible to characterize the connectivity of a graph with local graph traits, in the next section we focus on studying local graph traits which imply k-connectivity. In Section 5 we leverage these local graph traits to design local distributed algorithms.

4 Local Graph Traits That Imply k-Connectivity

We describe three local graph traits which imply graph k-connectivity for simply connected graphs. The first graph trait holds for general graphs, while the other two local traits hold only for unit disk graphs.

4.1 A Natural Local Trait for k-Connectivity

Perhaps the simplest and most intuitive local graph trait for k-connectivity is to check if the neighborhood of a vertex is k-connected. Specifically, consider the local trait $(c, K(k))$ where $c > 0$ is a positive constant and $K(k)$ is the predicate that checks if the c-neighborhood of a vertex is k-connected. We now show that this local graph trait implies k-connectivity.

Theorem 3. *The local graph trait $(c, K(k))$ implies k-connectivity for simply connected graphs.*

Proof. Suppose by contradiction that a connected graph satisfies the local graph trait $(c, K(k))$ but it is not k-connected. Since G satisfies $(c, K(k))$, then $G^c(u)$ must have at least $k+1$ vertices, hence $|V| \geq k+1$. Since by assumption G is not k-connected, it has a vertex cut with at most $k-1$ vertices. On the other hand since G is connected, any vertex cut is of size at least 1.

In particular let C denote a minimum vertex cut, and let P and Q be two connected components produced by removing all vertices in C. Fix any vertex $u \in C$, we make the following claim (proved later):

Claim. There exists vertices $p, q \in N(u)$ such that $p \in P$ and $q \in Q$.

Let $U = N^t[u] \setminus \{p, q\}$, where p and q are fixed as in the claim. Observe that since G satisfies $(c, K(k))$ then $G^c(u)$ is k-connected. In particular this means $|N^c[u]| \geq k+1$ and hence $|U| \geq k-1$. Moreover, this also implies that removing any subset of U of size at most $k-1$ leaves a path from p to q in $G^c(u)$.

However by assumption, removing the set $C \subseteq V$ of size at most $k-1$ produces two connected components P and Q. Since removing $C \subseteq V$ disconnects P from Q in G, then removing $U \cap C \subseteq U$ has to disconnect p and q in $G^c(u)$.

Finally since $|C| \leq k-1$ then $|U \cap C| \leq k-1$, but this contradicts that removing any subset of U of size at most $k-1$ leaves a path from p to q in $G^c(u)$, which completes the theorem. □

Proof. [of Claim 1] By assumption, C is a minimum vertex cut that separates P and Q. Hence, if we consider the smaller vertex set $C' = C \setminus \{u\}$, it must be that removing the vertices from C' does not separate P and Q.

This implies that for any pair of vertices $p' \in P$ and $q' \in Q$ there exists a simple path between p' and q' using only vertices from the set $V - C'$. Since this path does not exist when removing the set C, the path must go through u.

Follow the path starting at $p' \in P$, and let $p \in P$ be the last vertex in the path that belongs to P. It must be that the vertex in the path after p is u (and hence $p \in N(u)$). Otherwise it would contradict that P is a component separated from the rest of the vertices when removing C. By following the path starting at $q' \in Q$ the same argument can be used to show there exists a vertex $q \in Q$ such that $q \in N(u)$, which completes the proof. □

It's not immediate how to improve the accuracy of $(c, K(k))$. To illustrate this difficulty, assume there exists some local graph trait $(c, K'(k))$ which implies k-connectivity and is more accurate than $(c, K(k))$. Therefore, there must exist a k-connected graph G with a vertex $u \in V$ whose c-neighborhood does not satisfy $K(k)$ but does satisfy $K'(k)$. However, this also implies that if we consider the graph $G' = G^c(u)$, then G' is not k-connected but $K'(k)$ is satisfied at node u.

Using this logic it is tempting to go further and argue that since any local graph trait which implies k-connectivity should not be satisfied by a graph which is not k-connected, then $K'(k)$ does not imply k-connectivity (reaching a contradiction). However, a graph trait $(c, K'(k))$ is only satisfied by a graph, if the c-neighborhood of *all* nodes satisfies $K'(k)$.

The next subsection describes a local graph traits which is less accurate than (c, K). However, this local trait will introduce ideas which will inspire a better

graph trait presented in the last subsection, one which uses techniques that allow us to use it to preserve k-connectivity in Section 5. We remark that up to this point, we have not used either labeled or Euclidean graphs.

4.2 Small Edges Increase Connectivity

Given an Euclidean graph (G, p), we define the length of an edge as the Euclidean distance between the embedding of its endpoints. In unit disk graphs, one would expect that moving all nodes closer together (thereby decreasing the length of all the edges) would increase the connectivity of the graph. This observation is exploited by the local graph trait $(c, Small(k))$. Here $c > 0$ is a positive constant and $Small(k)$ is a predicate that checks if the c-neighborhood of a node has at least $k + 1$ nodes and has a connected spanning subgraph using only edges of length at most $1/k$. To prove that this local graph trait implies k-connectivity we first show the following:

Lemma 4. *If a unit disk graph with $n \geq k+1$ vertices has a connected spanning subgraph using edges of length at most $1/k$, then it is k-connected.*

Proof. Fix any unit disk graph graph G with $n \geq k + 1$ vertices which has a connected spanning graph using edges of length at most $1/k$. If G were a clique then it is k-connected, hence we assume G is not a clique and let C be a minimum vertex cut of G. We will show that $|C| \geq k$, which implies that G is k-connected.

Let P and Q be two connected components produced by the cut C. Since G has a connected spanning subgraph using edges of length at most $1/k$, then for any pair of vertices $p \in P$ and $q \in Q$ there exists a simple path $p \rightsquigarrow q$ from p to q in using only edges of length $1/k$. We use the vertices of C to define a *gap* in $p \rightsquigarrow q$, as a maximal set of contiguous vertices in $p \rightsquigarrow q$ that belong to C. For each gap g let $g.first$ and $g.last$ be the vertices in the path immediately before and after the gap.

Any gap g is of size at most $|g| \leq |C|$, and the Euclidean distance between $g.first$ and $g.last$ is bounded by $(|g| + 1)/k$. Hence if $|C| \leq k - 1$, then the distance between the $g.first$ and $g.last$ is at most $k/k = 1$. However, since G is a unit disk graph by assumption, there must exist an edge $(g.first, g.last)$ in G which bridges the gap and there would exist a path from p to q. Therefore, for C to be a cut, it must be that $|C| \geq k$, and thus G is k-connected. □

We can stitch Lemma 4 with Theorem 3, which showed that $(c, K(k))$ implies k-connectivity of connected graphs, to prove the following.

Theorem 5. *The local graph trait $(c, Small(k))$ implies k-connectivity for simply connected unit disk graphs.*

Proof. By Theorem 3 it suffices to show that, for every vertex $u \in V$, if the c-neighborhood of u satisfies $Small(k)$ then it also satisfies $K(k)$.

Fix a vertex $u \in V$ which satisfies $Small(k)$, then it follows that $G^c(u)$ has at least $k+1$ vertices and has a connected spanning subgraph using edges of length

at most $1/k$. However, then by Lemma 4 $G^c(u)$ is k-connected and it satisfies $K(k)$. □

In the process of proving Theorem 5 we showed that the graphs which satisfy $(c, Small(k))$ also satisfy $(c, K(k))$. It is not difficult to construct unit disk graphs which satisfy $(c, K(k))$ but where $(c, Small(k))$ does not hold (i.e. a clique with $k+1$ vertices using only "large" edges). Therefore it follows that by definition $(c, K(k))$ is more accurate than $(c, Small(k))$ for k-connectivity.

A natural question, is to ask weather *all* edges in the connected spanning graph need to be small for the connectivity of a graph to increase, or is it sufficient only for *some* edges to be small? We answer this question in the next subsection.

4.3 Spanning Trees with Small Edges Imply k-Connectivity

Given an Euclidean graph (G, p), let MST_G denote a minimum spanning tree of G. Observe that in labeled graphs, ties between edges of the same length can be broken consistently using the unique identifiers associated with each node. Therefore, in Euclidean labeled graphs, we can assume distinct edge lengths, which implies there is a unique minimum spanning tree.

For any positive constant $c > 0$, let $LMST_G^c = (V, F)$ denote the local minimum spanning tree of $G = (V, E)$. The edge set of the local minimum spanning tree is $F := \big\{\{u, v\} \mid \{u, v\} \in E(MST_{G^c(u)}) \cap E(MST_{G^c(v)})\big\}$. In other words, $LMST_G^c$ is the intersection of the minimum spanning trees associated with the c-neighborhood of every node in G. It is known [10] that in graphs G with a unique minimum spanning tree MST_G, the local minimum spanning tree contains the minimum spanning tree and is therefore connected. This property suggests an improved local graph trait using the same ideas of $(c, Small(k))$.

Consider the local graph trait $(c, MSTSmall(k))$ where $c > 0$ is any positive constant and $MSTSmall(k)$ is a predicate that checks if the c-neighborhood of a node u has at least $k+1$ nodes and all the edges of the form $\{u, v\}$ in its minimum spanning tree have length at most $1/k$. The next theorem shows that $(c, MSTSmall(k))$ implies k-connectivity as a consequence from the properties of $LMST_G^c$ and Lemma 4.

Theorem 6. *The local graph trait $(c, MSTSmall(k))$ implies k-connectivity for simply connected labeled unit disk graphs.*

Proof. Let G be any simply connected labeled unit disk graph. Therefore G has a unique minimum spanning tree MST_G. If G satisfies $(c, MSTSmall(k))$ then by definition it follows that all the edges in $LMST_G^c$ are of length $1/k$.

Moreover, since $MST_G \subset LMST_G^c$ and MST_G is a connected spanning graph by definition, then clearly $LMST_G$ is also a connected spanning subgraph of G. Finally since G has a connected spanning subgraph with edges of length at most $1/k$ (namely $LMST_G^c$), then Lemma 4 implies it is k-connected. □

A feature which is shared by graphs that satisfy $(c, MSTSmall(k))$ and $(c, Small(k))$, is that they contain connected spanning subgraphs using edges of length at most $1/k$. This will prove to be a valuable property in Section 6.

A well known folklore result is that amongst all spanning trees, a minimum spanning tree minimizes the length of the longest edge. Together with the fact that it is possible to construct a k-connected graph where $(c, MSTSmall(k))$ is satisfied, but not $(c, Small(k))$, we can conclude $(c, MSTSmall(k))$ is more accurate than $(c, Small(k))$ with respect to k-connectivity.

In fact, it turns out that $(c, MSTSmall(k))$ is satisfied in some k-connected graphs where $(c, K(k))$ is *not* satisfied. However, the converse is also true, and therefore the accuracy of $(c, K(k))$ and $(c, MSTSmall(k))$ for k-connected graphs is incomparable. Which of them is more useful depends on the characteristics of the graphs being considered.

5 Applying Local Graph Traits to Distributed Algorithms

As a warm up, we first consider the simplest applications of the local graph traits described in Section 3. Specifically, we use them to design a local distributed algorithm to test for a k-connected graph.

Consider the following constant time procedure (which is the algorithmic counterpart of the local trait $(c, K(k))$). The process running at each node $u \in V(G)$ executes a full information protocol for $c + 1$ communication rounds to recover the c-neighborhood of u. At the end of the $c + 1$ rounds, the process outputs true if $G^c(u)$ is k-connected and outputs false otherwise.

Since $(c, K(k))$ implies k-connectivity, then if this procedure outputs true at every node, then G is guaranteed k-connected. On the other hand, if G is not k-connected we are guaranteed that at least one process will output false.

This procedure can be used by itself as a constant time distributed algorithm to test for k-connectivity, or can be used as a building block to solve other problems. For example in a distributed topology control algorithm, to guarantee k-connectivity, every process could run the procedure repeatedly with an increasing power assignment, stopping when the procedure outputs true. If the maximum transmission power is sufficiently large and the graph has at least $k+1$ nodes, this algorithm eventually stops and guarantee a k-connected graph. However, there is no guarantee that it will stop in the first round when the graph becomes k-connected. Moreover, the impossibility result on weakly-local graph traits for k-connectivity, implies that any distributed topology control algorithm that finds an optimal solution requires at least $\Omega(n)$ communication rounds.

In deployments where the unit disk graph assumption holds and nodes are equipped with GPS, an algorithmic implementation of $(c, MSTSmall(k))$ requires one less communication round, and might yields better results.

In the remainder of this section we discuss another application of local graph traits. Namely, we show how to leverage the local graph trait $(c, MSTSmall(k))$ together with the connectivity preserving algorithm presented in [4], to yield a k-connectivity preserving algorithm for mobile ad hoc networks.

5.1 Maintaining k-Connectivity of Robot Swarms

We consider a mobile ad hoc network composed of n mobile robots (aka processes). When possible we adhere to the standard synchronous network model described in Section 2. At the beginning of every round, in addition to the usual operations, each robot can query its own position (perhaps using GPS), query its intended target position for the next round (via an existing motion planner) and feed a trajectory to its actuators (for example, a linear trajectory to its intended target). Actuators are imperfect, and hence a robot following a trajectory may stop or slow down abruptly and travel only a fraction, possible none, of this trajectory. We assume the communication graph is a unit disk graph induced by the positions of the robots. For simplicity, we will assume that at the beginning of every round each robot knows its neighbors in the communication graph and their positions, this could be implemented by exchanging hello messages tagged with the position of the robots.

Since robots (as opposed to regular processes) can move and change their position from round to round, we extend our notation to account for this. Let $p(v, r)$ denote the position of the robot occupying node v at round r. Similarly, let $G(r) = (V, E(r))$ denote the communication graph induced at round r, and let $N[u, r]^t$ be the closed t-neighbors of node u at round r. We use $N[u, r]$ as short hand notation for the closed 1-neighbors of u at round r.

In previous work [2, 4] we addressed the problem of maintaining connectivity ($k = 1$) for robot swarms. Specifically, we described a distributed algorithm that modifies an existing short-term motion plan to ensure connectivity. The algorithm uses only local information, is stateless, does not require a fixed set of neighbors and does not make any assumptions on the current or goal configurations. Moreover, the algorithm is *robust* to the robots' speed changes; if robots travel any fraction of the trajectory (perhaps none) at any speed, connectivity is preserved. The *progress* of the algorithm is defined as the total distance traveled by all robots (summing over all the robots) towards their intended destinations. Let d be the minimal distance each robot intends to move and let R be the communication radius. Assuming that the target configuration of the robots is connected and the motion does not require breaking any cycles, we proved that the algorithm guarantees that the progress is at least $\min(d, R)$. Furthermore, we exhibited a class of configurations where no local algorithm can do better than this bound, and hence under these conditions the bound is tight and the algorithm is asymptotically optimal. Finally we proved that all robots get ε-close to their target within $O(D_0/R + n^2/\varepsilon)$ rounds where D_0 is the total initial distance to the targets and n is the number of robots [4].

Starting with a graph which satisfies the local trait $(c, MSTSmall(k))$, we describe how to extend the ConnServ algorithm presented in [4] to enforce the local graph trait $(c, MSTSmall(k))$ throughout the execution and preserve k-connectivity with similar robustness, safety and progress conditions as the original algorithm.

Connectivity Maintenance Algorithm. The ConnServ algorithm [4] is parametrized by a communication radius $R \in \mathbb{R}$ and a neighbor filtering function

$f : 2^V \to 2^V$ which receives a closed set of neighbors $N[u, r]$ and returns a filtered set of neighbors $N'(u, r) \subseteq N[u, r]$. These parameters should satisfy the following properties: $P1$. Any two robots which are at distance R or less can reliably exchange a message (i.e. are connected). $P2$. Filtered neighbors are within distance R ($\forall v \in N[u, r]$, $\|p(v) - p(v)\| \leq R$). $P3$. Preserving connectivity with the filtered neighbors is sufficient to preserve global graph connectivity. Formally, if $G(r) = (V, E(r))$ is connected, then the spanning subgraph $H = (V, F)$ where $F := \{\{u, v\} \mid u \in N'(v, r) \wedge v \in N'(u, r)\}$ is also connected.

When run by a robot at node u at round r, the input of the ConnServ algorithm is a tuple (p_u, N_u, t_u), where $p_u = p(u, r)$, $N_u = N[u, r]$ and t_u is the intended target position at round r. The output of the ConnServ algorithm is a new target position t^*_u. For any parameters which satisfy the properties above, the ConnServ algorithm was shown to provide the following guarantees [4].

Safety Theorem. *If $u \in N'(v, r)$ and $v \in N'(u, r)$, then $|t^*_u - t^*_v| \leq R$*

In other words, if by the beginning of the next round every robot moves to the target position output by ConnServ , if $G(r)$ was connected then $G(r + 1)$ will also be connected (this follows from the safety theorem and the guarantees assumed on the filtering function).

However, it would be unreasonable to expect all robots to be able to reach the target output by the algorithm by the beginning of the next round. For example, a robot might encounter an obstacle, it might stop or slow down suddenly due to hardware malfunction, or it might be to slow to complete the trajectory. This motivates the next result, which shows that the graph will remain connected even if robots stop or slow down unexpectedly.

Robustness Theorem. *If $u \in N'(v, r)$ and $v \in N'(u, r)$, then for any point p in the linear trajectory from $p(u, r)$ to t^*_u, and any point q in the linear trajectory from $p(v, r)$ to t^*_v, it holds that $\|p - q\| \leq R$.*

For the algorithm to be useful, it needs to provide a progress guarantee that relates the input and the output target, since a trivial algorithm which forces all robots to remain stationary vacuously provides the previous safety and robustness properties. On the other hand, it is not possible to guarantee progress unconditionally, since, for example, if two robots want to move in opposite directions as to disconnect the graph, guaranteeing any progress would violate the safety and robustness properties. Therefore, our progress guarantees are conditioned on the assumption that the intended targets do not require breaking any edges needed for connectivity.

We define the progress of a robot as the distance advanced to the input target assuming it moves to the output target. If at round r a robot at node u has an input target t_u, let $d_u = \|p(u, r) - t_u\|$ be the distance from its current position to its input target. If the algorithm ConnServ outputs a target t^*_u the progress is defined as $\delta_u = d_u - \|t_u - t^*_u\|$

The progress of the system is then the sum of the progress of each robot, that is $\sum_{u \in V} \delta_u$. In the following d is defined as $d = \min_{u \in V} d_u$.

Progress Theorem. *In any configuration where the intended targets do not require breaking edges needed for connectivity, the progress is at least* $\min(d, R)$.

Finally, assuming the robots have the same target for sufficiently many rounds, the following result provides an upper bound on the number of rounds required for every robot to reach their target. Here $D_0 = \sum_{u \in V} d_u$ is the sum of the distances from each robot to its intended long term target.

Termination Theorem. *In any configuration where the intended targets do not require breaking edges needed for connectivity, every robot gets ε-close to its target within $O(D_0/R + n^2/\varepsilon)$ rounds.*

k-Connectivity Maintenance. We will argue that if the starting configuration satisfies $(c, MSTSmall(k))$, it is possible to select the communication radius R and the filtering function f to guarantee the ConnServ algorithm preserves k-connectivity.

Concretely, we let $R = R_{\min}/k$ where $R_{\min}$ is the smallest distance such that any two robots within distance $R_{\min}$ can exchange messages reliably. For the filtering function f, let $S \subseteq N[u, r]$ be the subset of vertices which are at distance less than or equal to $R_{\min}/k$ from u. We let f return every vertex v such that $v \in E(MST_S)$, in other words the neighbors of u in the minimum spanning tree involving only vertices in S (i.e. closer than $R_{\min}/k$ to u). Finally, we assume the communication graph is initially k-connected, or more specifically, we assume $G(0)$ satisfies the local graph trait $(c, MSTSmall(k))$.

The parameters described satisfy $P1$ and $P2$, however it is not evident that the filtering defined satisfies $P3$ and much less that the resulting algorithm guarantees k-connectivity.

Theorem 7. *$\forall r \geq 0$, f satisfies $P3$ and $G(r)$ is k-connected.*

Proof. Let $H(r)$ be the graph that results from removing all edges of $G(r)$ which are of length more than $R_{\min}/k$. We make the following claim (proved later).

Claim. $H(r)$ is a *connected* spanning subgraph of $G(r)$.

Then it follows that $H(r)$ satisfies $(c, MSTSmall(k))$ and hence $H(r)$ (and therefore $G(r)$) are k-connected. Finally since the filtered neighbors returned by f define a local minimum spanning tree over $H(r)$, f satisfies $P3$. □

Proof. [of Claim] We proceed by induction on r. The base case is trivial since $G(0)$ satisfies $(c, MSTSmall(k))$. Suppose by inductive hypothesis that $H(r)$ is a connected spanning subgraph of $G(r)$. Let $LMST(r)$ be the connected spanning subgraph of $H(r)$ described by the filtered neighbors returned by f.

By the safety and robustness theorems, all the edges of $LMST(r)$ are present in $G(r+1)$ with length at most $R = R_{\min}/k$. Therefore the subgraph $H(r+1) \subseteq G(r+1)$ also contains $LMST(r)$, and thus it is a connected spanning subgraph of $G(r+1)$. □

Therefore, since the parameters chose for the ConnServ algorithm satisfy $P1, P2$ and $P3$, the safety and robustness theorems imply the graph is

k-connected at every time instant even if the robots slow down or stop unexpectedly and only execute some fraction of the trajectory.

On the other hand, the progress and termination theorems imply at every round the progress of the system is at least $\min(d, R_{\min}/k)$ and the system becomes ε-close to its targets within $O(D_0 k/R_{\min} + n^2/\varepsilon)$. Therefore, preserving k-connectivity via this extension to the ConnServ algorithm incurs in a cost linear in k compared to preserving simple connectivity.

References

[1] B. Awerbuch, M. Luby, AV Goldberg, and S.A. Plotkin. Network decomposition and locality in distributed computation. *Proc. of the 30th Annual Symposium on Foundations of Computer Science*, 1989.

[2] A. Cornejo and N. Lynch. Connectivity Service for Mobile Ad-Hoc Networks. *Spatial Computing Workshop*, 2008.

[3] A. Cornejo and N. Lynch. Fault-Tolerance Through k-Connectivity. *Workshop on Network Science and Systems Issues in Multi-Robot Autonomy: ICRA 2010*, 2, 2010.

[4] A. Cornejo, F. Kuhn, R. Ley-Wild, and N. Lynch. Keeping mobile robot swarms connected. *DISC 2009: 23rd International Symposium on Distributed Computing*, September 23-25 2009.

[5] A. Czumaj and H. Zhao. Fault-tolerant geometric spanners. *Discrete and Computational Geometry*, 32(2):207–230, 2004.

[6] G. A. Dirac. Some theorems on abstract graphs. *Proc. London Mathematical Society*, 2, 1952.

[7] X. Jia, D. Kim, S. Makki, P.J. Wan, and C.W. Yi. Power assignment for k-connectivity in wireless ad hoc networks. *Journal of Combinatorial Optimization*, 9(2):213–222, 2005.

[8] M. Jorgic, N. Goel, K. Kalaichevan, A. Nayak, and I. Stojmenovic. Localized detection of k-connectivity in wireless ad hoc, actuator and sensor networks. *Proc. 16th ICCCN*, 2007.

[9] N. Li and J.C. Hou. FLSS: a fault-tolerant topology control algorithm for wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 275–286. ACM New York, NY, USA, 2004.

[10] N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. *INFOCOM*, 3:1702–1712, 2003.

[11] N. Linial. Distributive graph algorithms Global solutions from local data. In *28th Annual Symposium on Foundations of Computer Science, 1987.*, pages 331–335, 1987.

[12] R. Thurimella. Sub-linear distributed algorithms for sparse certificates and biconnected components. In *PODC*, pages 28–37. ACM New York, NY, USA, 1995.

[13] A. Wigderson. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM (JACM)*, 30(4):735, 1983.